

# Code ESP32

Ce projet contient le code source pour le microcontrôleur ESP32 utilisé pour détecter des balises BLE (Beacon) et transmettre les données via MQTT à un serveur.

## Description

Le code permet de :

- Se connecter à un réseau Wi-Fi.
- Effectuer un scan BLE (Bluetooth Low Energy) pour détecter des balises compatibles.
- Extraire et formater les données détectées :
  - **idSTRI** : UUID de la balise (16 octets à partir de l'index 2).
  - **année** : Valeur major extraite (2 octets). (index 16-17).
  - **idBadge** : Valeur minor extraite (2 octets).(index 18-19).
  - **mac\_address\_detectee** : Adresse MAC du périphérique détecté.(index 22-27).
  - **macBLE** : Adresse MAC BLE du détecteur ESP32.
  - **macWIFI** : Adresse MAC Wi-Fi du détecteur ESP32.
- Envoyer les données formatées sous forme de JSON sur des topics MQTT.

## Exemple de données envoyées

```
{  
  "idSTRI": "2D7A9F0C-E0E8-4CC9-A71B-A21DB2D034A1",  
  "année": 2025,  
  "idBadge": 222,  
  "mac_address_detectee": "DF:28:82:32:2D:F3",  
  "macBLE": "C8:C9:A3:CC:1B:C6",  
  "macWIFI": "C8:C9:A3:CC:1B:C4"  
}
```

## Pré-requis

- **Matériel** : ESP32
- **Logiciel** :
  - Arduino IDE
- **Bibliothèques nécessaires** :
  - **WiFi** : Pour la connexion au réseau Wi-Fi.
  - **PubSubClient** : Pour la communication MQTT.
  - **ArduinoJson** : Pour la création de messages JSON.

- **BLEDevice** : Pour le scan BLE et la gestion des balises.

## Installation

### 1. Configuration matérielle :

- Connectez votre ESP32 à votre ordinateur.

### 2. Installation des bibliothèques nécessaires :

- Dans Arduino IDE, accédez à **Outils > Gérer les bibliothèques** et installez les bibliothèques suivantes :
  - **WiFi**
  - **PubSubClient**
  - **ArduinoJson**
  - **BLEDevice**

**Configuration Wi-Fi** : Dans le code, remplacez les valeurs des constantes par vos propres identifiants :

```
const char* ssid = "Votre_SSID";      // Nom de votre réseau Wi-Fi
const char* password = "Votre_MotDePasse"; // Mot de passe de votre réseau Wi-Fi
```

3.

**Configuration du serveur MQTT** : Configurez l'adresse IP et le port du serveur MQTT :

```
const char* mqttServer = "x.x.x.x"; // Adresse IP du serveur MQTT
const int mqttPort = 1883;          // Port du serveur MQTT
const char* mqttUser = "admin";     // Identifiant MQTT
const char* mqttPassword = "xhc9QmmISs"; // Mot de passe MQTT
```

### 4. Configuration des partitions mémoire :

- Dans Arduino IDE, accédez à **Outils > Partition Scheme** et sélectionnez : **2MB APP / 2MB SPIFFS**.

### 5. Téléversement :

- Compilez et téléversez le code sur l'ESP32.

### 6. Vérification des logs :

- Ouvrez le moniteur série (115200 baud) pour suivre les messages de débogage.

# Topics MQTT utilisés

## 1. Topic 1 : /detecteur/porte/macwifi/<macwifi>

- **Description** : Ce topic est utilisé pour publier les données détectées, spécifiques à la MAC Wi-Fi du détecteur.

## 2. Topic 2 : /detecteur/porte/iddetecteur/x

- **Description** : Ce topic est utilisé pour publier les mêmes données détectées, spécifiques à l'identifiant constant du détecteur (x).
- **Remarque** : **La valeur de x est une constante à configurer dans le code.**

Exemple de publication :

```
{
  "idSTRI": "2D7A9F0C-E0E8-4CC9-A71B-A21DB2D034A1",
  "année": 2025,
  "idBadge": 222,
  "mac_address_detectee": "DF:28:82:32:2D:F3",
  "macBLE": "C8:C9:A3:CC:1B:C6",
  "macWIFI": "C8:C9:A3:CC:1B:C4"
}
```

```
stri@raspberrypi:~$ mosquitto_sub -h localhost -u admin -P xhc9Qm1Ss -p 1883 -t "/detecteur/porte/iddetecteur/1"
{"idSTRI": "2D7A9F0C-E0E8-4CC9-A71B-A21DB2D034A1", "année": 2025, "idBadge": 222, "mac_address_detectee": "DF:28:82:32:2D:F3", "macBLE": "C8:C9:A3:CC:1B:C6", "macWIFI": "C8:C9:A3:CC:1B:C4"}
stri@raspberrypi:~$ mosquitto_sub -h localhost -u admin -P xhc9Qm1Ss -p 1883 -t "/detecteur/porte/macwifi/C8:C9:A3:CC:1B:C4"
{"idSTRI": "2D7A9F0C-E0E8-4CC9-A71B-A21DB2D034A1", "année": 2025, "idBadge": 222, "mac_address_detectee": "DF:28:82:32:2D:F3", "macBLE": "C8:C9:A3:CC:1B:C6", "macWIFI": "C8:C9:A3:CC:1B:C4"}
```

## Signification des données envoyées

Champ	Description
idSTRI	UUID unique de la balise détectée.(id de l'STRI)
année	Champ major : représente l'année ( définie par l'utilisateur).
idBadge	Champ minor : identifiant de l'étudiant.
mac_address_detectee	Adresse MAC de la balise BLE détectée.

macBLE	Adresse MAC BLE du détecteur ESP32.
macWIFI	Adresse MAC Wi-Fi du détecteur ESP32.